**Exercise 2: Implementing Dependency Injection**

**Scenario:**

In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI.

**Steps:**

1. **Modify the XML Configuration:**
   * Update **applicationContext.xml** to wire **BookRepository** into **BookService**.

**applicationContext.xml :**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- BookRepository Bean -->

<bean id="bookRepository" class="com.library.repository.BookRepository" />

<!-- BookService Bean with DI -->

<bean id="bookService" class="com.library.service.BookService">

<property name="bookRepository" ref="bookRepository" />

</bean>

</beans>

1. **Update the BookService Class:**
   * Ensure that **BookService** class has a setter method for **BookRepository**.

**BookService.java :**

package com.library.service;

import java.util.List;

import org.springframework.stereotype.Service;

import com.library.model.Book;

import com.library.repository.BookRepository;

*@Service*

public class BookService {

private BookRepository bookRepository;

// Setter for dependency injection (Spring will call this method)

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void showBooks() {

List<Book> books = bookRepository.getAllBooks();

if (books.isEmpty()) {

System.***out***.println("No books found.");

} else {

System.***out***.println("Available Books:");

for (Book book : books) {

System.***out***.println(book);

}

}

}

}

**BookRepository.java :**

package com.library.repository;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Repository;

import com.library.model.Book;

*@Repository*

public class BookRepository {

// Simulated database using a list

private List<Book> bookList;

public BookRepository() {

bookList = new ArrayList<>();

bookList.add(new Book(1, "Wings of Fire", "Dr. A.P.J. Abdul Kalam"));

bookList.add(new Book(2, "Gitanjali", "Rabindranath Tagore"));

bookList.add(new Book(3, "How to Talk to Anyone", "McGraw-Hill"));

}

// Simulates fetching data from a DB

public List<Book> getAllBooks() {

return bookList;

}

}

3.**Test the Configuration:**

* + Run the **LibraryManagementApplication** main class to verify the dependency injection.

**LibraryManagementApplication,java :**

packagepackage com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class MainApp {

public static void main(String[] args) {

// Load Spring application context

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

// Retrieve the bean with dependency injected

BookService bookService = (BookService) context.getBean("bookService");

// Verify that BookRepository was injected

System.***out***.println("===== Dependency Injection Test =====");

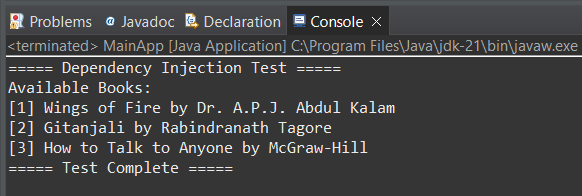
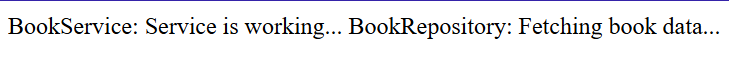
bookService.showBooks();

System.***out***.println("===== Test Complete =====");

}

}

**OUTPUT:**

****